

# Einführung in die wissenschaftliche Methodik der Informatik (01604)

## Einsendeaufgaben zur Kurseinheit 3

### Aufgabe 2.1 — 8 Punkte.

Beschaffen Sie weitere bibliographische Informationen über

- den Artikel „An Impossibility Theorem for Clustering“ von Jon Kleinberg,
- das Buchkapitel „Motion Planning“ von Lydia Kavraki und Steven LaValle.

- a) Erstellen Sie für beide Werke jeweils eine vollständige Quellenangabe.
- b) Finden Sie zwei Werke, die mindestens eines der Werke zitieren, und erstellen Sie auch für diese Werke jeweils eine vollständige Quellenangabe.

### Aufgabe 2.2 — 4 Punkte.

Schreiben Sie den folgenden Satz zweimal ab und formatieren Sie die Verweise dabei einmal in klassischer Havard-Zitierweise und einmal in der Zitierweise mit Namenskürzeln.

„Interaktive Beweissysteme (vgl. [1]) bilden die Grundlage der Komplexitätsklasse  $IP$ , welche der Klasse  $PSPACE$  entspricht [2].“

[1] Shafi Goldwasser, Silvio Micali und Charles Rackoff: *The knowledge complexity of interactive proof systems*. SIAM Journal on Computing 18(1), 1989, Seiten 186–208.

[2] Adi Shamir:  $IP = PSPACE$ . Journal of the Association for Computing Machinery 39(4), 1992, Seiten 869–877.

### Aufgabe 2.3 — 10 Punkte.

Im Internet findet sich ein Artikel mit dem Titel „Open Problems on Polytope Reconstruction“ von Erik D. Demaine und Jeff Erickson. Diskutieren Sie, ob dieses Werk zitierfähig oder zitierwürdig ist.

### Aufgabe 2.4 — 10 Punkte.

Unten ist der Anfang einer fiktiven Einleitung abgedruckt. Diskutieren Sie den Schreibstil dieser Einleitung. Analysieren Sie dafür, bezogen auf Kapitel 3.2, jeden Absatz einzeln und geben Sie an, welche Regeln des wissenschaftlichen Schreibens/Arbeitens jeweils verletzt werden. Begründen Sie Ihre Einschätzungen kurz.

Finden Sie zudem zwei Regeln, die in mindestens zwei Absätzen eingehalten werden.

## Mergesort mit fehlerbehafteten Daten

In der heutigen Zeit, in der Algorithmen von vielen Akteuren als das reine Böse angesehen und verdammt werden, ist es um so wichtiger die Höhepunkte und Glanzleistungen aus der Welt der Algorithmen und Ihrer Erfinder zu würdigen. In dieser Arbeit beleuchte ich für Sie, liebe Leserinnen, verschiedene Teilspekte des unbestritten elegantesten Sortieralgorithmus den es gibt, nämlich Mergesort. Basierend auf dem simplen und dennoch genialen Prinzip des *Teile und Herrsche* greifen die einzelnen Bestandteile dieses Algorithmus absolut geräuschlos ineinander und entwickeln dennoch eine Spannung, die ich nur selten erlebt habe.

Dies liegt nicht nur, obwohl man diesen Zusammenhang auch nicht leugnen kann, vordergründig an den offensichtlichen und naturgegebenen Gegensätzen, die in diesem Algorithmus Anwendung finden, sprich, zwischen den Prinzipien des Sortierens, die Ordnung erzeugen, und den Prinzipien des Mischens, die Unordnung erzeugen, denn was könnte unterschiedlicher, ja konfliktbeladener sein.

Zunächst zur präzisen Problembeschreibung: Gegeben sind gemischte Daten, welche (möglichst zügig, verlustfrei und ohne Beschädigungen) in eine natürliche Rangfolge gebracht werden müssen. Mit Hinblick auf diese Aufgabenstellung, wird in dieser Arbeit das Verhalten des Algorithmus für fehlerbehaftete Datenmengen untersucht. Dabei soll insbesondere festgestellt werden, ob das *Teilen* oder das *Herrschen* den größeren Einfluss auf die Ergebnisse hat. Es soll konkret geprüft werden, ob bei fehlerbehafteten Daten, die Funktion des Algorithmus garantiert wird. Wenn ja, soll die Frage untersucht werden, ob die Laufzeitschranken trotzdem mehr als 100%ig eingehalten werden.

Der Algorithmus ist schnell beschrieben. Zunächst scannt dieser die Werte und verteilt ebenjene auf genau zwei Listen. Auf diese Weise enthält jeder Vektor, eventuell können es manchmal auch mehr als zwei sein, also maximal  $\theta$  Daten. Dieser Schritt ist trivial und entspricht dem *Teilen*. Das *Herrschen* besteht daraus, die Datenfelder rekursiv zu sortieren, und dann fachgerecht zusammenzumischen.

### Aufgabe 2.5 — 8 Punkte.

Nehmen Sie eine Auswertung der in den Tabellen aufgeführten Daten vor.

- Stellen Sie die in Tabelle 2.1 dargestellten Daten mit Hilfe eines geeigneten Diagramms dar.
- Berechnen Sie für die in Tabelle 2.2 genannten Algorithmen jeweils Mittelwert, Median und Standardabweichung der fiktiven Laufzeiten.
- Berechnen Sie Mittelwert, Median und Standardabweichung über alle in Tabelle 2.2 aufgeführten Daten.

Wort	The Fellowship of the Ring	The Two Towers	The Return of the King
away	335	341	258
dark	413	380	300
great	398	422	481
hobbit	410	260	132
lord	68	114	265
mordor	62	68	93
mountain	172	136	129
precious	14	92	21
ring	392	98	105
sword	36	51	44

Tabelle 2.1: Absolute Häufigkeiten einiger Worte in „Der Herr der Ringe“.

Algorithmus	System 1	System 2	System 3	System 4	System 5	System 6
Mergesort	0.0123 ms	0.0212 ms	0.0162 ms	0.0113 ms	0.2311 ms	0.0502 ms
Bubblesort	0.0222 ms	0.0441 ms	0.0360 ms	0.0452 ms	0.9142 ms	0.3320 ms
Quicksort	0.0133 ms	0.0142 ms	0.0161 ms	0.0093 ms	0.2113 ms	0.0492 ms

Tabelle 2.2: Fiktive Laufzeiten von Sortieralgorithmen auf verschiedenen Systemen.